IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	Date: April 6, 2006
Tanya COUCH, et al.	Confirmation No. 6531
Serial No: 10/037,659	Group Art Unit: 2164
Filed: January 2, 2002	Examiner: Jacob F. BETIT

For: METHOD AND SYSTEM FOR CONVERTING MESSAGE DATA INTO RELATIONAL TABLE FORMAT

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBSTITUTE APPEAL BRIEF

Dear Sir or Madam:

In response to the Office Communication mailed March 16, 2006, Appellants submit this Substitute Appeal Brief pursuant to 37 C.F.R. § 41.37.

I.   REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corp. of Armonk, New York by virtue of an assignment from the inventors recorded in the U.S. Patent and Trademark Office on January 2, 2002, at Reel No. 012455, Frame No. 0382.

II.   RELATED APPEALS AND INTERFERENCES

There are no appeals, interferences, or judicial proceedings known to Appellants, the Appellants' legal representative, or Assignee, which may be related to, directly affect, be directly affected by, or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

III.   STATUS OF CLAIMS

Claims 1-90 stand rejected. Appeal is taken from rejection of all of the foregoing claims 1-90.

IV.   STATUS OF AMENDMENTS

No amendments were filed subsequent to the final Office action dated May 18, 2005.

V.   SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 recites a method for converting messaging data into a relational table format in a database system, wherein the messaging data is within a messaging system. The method includes providing a plurality of table formatting specifications. *See, e.g.*, pg. 10, lns. 13-17; pg. 12, ln. 10 to pg. 13, ln. 21; pg. 15, lns. 1-5; FIGs. 7, 8, 8A, and 8B. The method also includes utilizing the plurality of table formatting specifications to automatically build and store a table function in the database system. *See, e.g.*, pg. 9, lns. 10-13; pg. 10, lns. 13-17; pg. 11, ln. 18 to pg. 12, ln. 1; pg. 14, lns. 12-18; pg. 15, ln. 5; FIGs. 10 and 10A. In addition, the method includes invoking the table function from within the database system to access the messaging data (120). *See, e.g.*, pg. 9, lns. 20-22; pg. 15, lns. 6-10; FIG. 2. The method further includes

converting the messaging data by the table function into specific data types according to the plurality of table formatting specifications, wherein the messaging data is transformed into the relational table format (140). *See, e.g.,* pg. 9, lns. 10-17; pg. 10, lns. 10-11; pg. 15, lns. 6-13; FIG. 2.

Dependent claim 2 depends from claim 1 and recites that the table function invokes at least one messaging function within the database system. *See, e.g.,* pg. 9, lns. 10-14 and 20-22; pg. 15, lns. 6-8; FIGs. 1 and 2.

Dependent claim 3 depends from claim 2 and recites that the table function and the at least one messaging function are user-defined functions within the database system. *See, e.g.,* pg. 8, lns. 16-18; pg. 9, lns. 13-14 and 20-22; pg. 11, lns. 15-19; pg. 15, lns. 6-8; FIGs. 1 and 2.

Independent claim 27 recites a computer readable medium containing programming instructions for converting messaging data into a relational table format in a database system, wherein the messaging data is within a messaging system. The computer readable medium includes programming instructions for providing a plurality of table formatting specifications. *See, e.g.,* pg. 10, lns. 13-17; pg. 12, ln. 10 to pg. 13, ln. 21; pg. 15, lns. 1-5; FIGs. 7, 8, 8A, and 8B. The computer readable medium also includes programming instructions for utilizing the plurality of table formatting specifications to automatically build and store a table function in the database system. *See, e.g.,* pg. 9, lns. 10-13; pg. 10, lns. 13-17; pg. 11, ln. 18 to pg. 12, ln. 1; pg. 14, lns. 12-18; pg. 15, ln. 5; FIGs. 10 and 10A. In addition, the computer readable medium includes programming instructions for invoking the table function from within the database system to access the messaging data (120). *See, e.g.,* pg. 9, lns. 20-22; pg. 15, lns. 6-10; FIG. 2. The computer readable medium further includes programming instruction for converting the

messaging data by the table function into specific data types according to the plurality of table formatting specifications, wherein the messaging data is transformed into the relational table format (140). *See, e.g.*, pg. 9, lns. 10-17; pg. 10, lns. 10-11; pg. 15, lns. 6-13; FIG. 2.

Dependent claim 28 depends from claim 27 and recites that the table function invokes at least one messaging function within the database system. *See, e.g.*, pg. 9, lns. 10-14 and 20 22; pg. 15, lns. 6-8; FIGs. 1 and 2.

Dependent claim 29 depends from claim 28 and recites that the table function and the at least one messaging function are user-defined functions in the database system. *See, e.g.*, pg. 8, lns. 16-18; pg. 9, lns. 13-14 and 20-22; pg. 11, lns. 15-19; pg. 15, lns. 6-8; FIGs. 1 and 2.

Independent claim 53 recites system for converting messaging data into a relational table format in a database system, wherein the messaging data is within a messaging system. The system includes a processor (50). *See, e.g.*, pg. 8, lns. 5-18; FIG. 1. The system also includes a table function building application executable by the processor (50) for receiving a plurality of table formatting specifications and for utilizing the plurality of table formatting specifications to automatically build and store a table function in the database system. *See, e.g.*, pg. 9, lns. 4-13; pg. 10, lns. 13-17; pg. 11, ln. 18 to pg. 12, ln. 1; pg. 12, ln. 10 to pg. 13, ln. 21; pg. 14, lns. 12-18; pg. 15, lns. 1-5; FIGs. 1, 7, 8, 8A, 8B, 10, and 10A. In addition, the system includes means for invoking the table function from within the database system to access the messaging data (120), wherein once invoked, the table function converts the messaging data into specific data types according to the plurality of table formatting specifications and transforms the messaging data into the relational table format (140). *See, e.g.*, pg. 9, lns. 10-22; pg. 15, lns. 6-13; FIG. 2.

Dependent claim 54 depends from claim 53 and recites that the table function invokes at least one messaging function within the database system. *See, e.g.*, pg. 9, lns. 10-14 and 20 22; pg. 15, lns. 6-8; FIGs. 1 and 2.

Dependent claim 55 depends from claim 54 and recites that the table function and the at least one messaging function are user-defined functions within the database system. *See, e.g.*, pg. 8, lns. 16-18; pg. 9, lns. 13-14 and 20-22; pg. 11, lns. 15-19; pg. 15, lns. 6-8; FIGs. 1 and 2.

Independent claim 67 recites a system for generating a customized invocation mechanism. The system includes an interface for receiving customizations. *See, e.g.*, pg. 10, lns. 13-17; pg. 12, ln. 10 to pg. 13, ln. 21; pg. 15, lns. 1-5; FIGs. 7, 8, 8A, and 8B. The system also includes a software module coupled to the interface for building an invocation mechanism based on the customization specifications and storing the invocation mechanism in a database, wherein the invocation mechanism is invokable by the database for accessing data external to the database. *See, e.g.*, pg. 9, lns. 10-13; pg. 10, lns. 13 17; pg. 11, ln. 18 to pg. 12, ln. 1; pg. 14, lns. 12-18; pg. 15, ln. 5; FIGs. 1, 10, and 10A.

Independent claim 75 recites a method for generating a customized invocation mechanism. The method includes receiving customization specifications. *See, e.g.*, pg. 10, lns. 13-17; pg. 12, ln. 10 to pg. 13, ln. 21; pg. 15, lns. 1-5; FIGs. 7, 8, 8A, and 8B. The method also includes building an invocation mechanism based on the customization specifications and storing the invocation mechanism in a database, wherein the invocation mechanism is invokable by the database for accessing data external to the database. *See, e.g.*, pg. 9, lns. 10-13; pg. 10, lns. 13-17; pg. 11, ln. 18 to pg. 12, ln. 1; pg. 14, lns. 12-18; pg. 15, ln. 5; FIGs. 1, 10, and 10A.

Independent claim 83 recites a program product containing instructions executable by a computer, the instructions embodying a method for generating a customized invocation mechanism. The program product includes instructions for receiving customization specifications. *See, e.g.*, pg. 10, lns. 13-17; pg. 12, ln. 10 to pg. 13, ln. 21; pg. 15, lns. 1-5; FIGs. 7, 8, 8A, and 8B. The program product also includes instructions for building an invocation mechanism based on the customization specifications and storing the invocation mechanism in a database, wherein the invocation mechanism is invokable by the database for accessing data external to the database. *See, e.g.*, pg. 9, lns. 10-13; pg. 10, lns. 13 17; pg. 11, ln. 18 to pg. 12, ln. 1; pg. 14, lns. 12-18; pg. 15, ln. 5; FIGs. 1, 10, and 10A.

VI.   GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1.   Appellants request review as to claims 1-5, 10-12, 14-17, 22-24, 26-31, 36-38, 40-43, 48-50, 52-58, 64-65, and 67-90 and their rejection under 35 U.S.C. § 102(e) as being anticipated by U.S. Pat. App. Pub. No. 2002/0046248 to Drexler (hereinafter "Drexler").

2.   Appellants request review as to claims 6-9, 32-35, and 59-63 and their rejection under 35 U.S.C. § 103(a) as being unpatentable over Drexler in view of U.S. Pat. No. 5,870,761 to Demers et al. (hereinafter "Demers").

3.   Appellants request review as to claims 13 and 39 and their rejection under 35 U.S.C § 103(a) as being unpatentable over Drexler in view of U.S. Pat. No. 6,704,742 to Huth et al. (hereinafter "Huth").

4.   Appellants request review as to claims 18-21, 25, 44-47, 51, and 66 and their rejection under 35 U.S.C. § 103(a) as being unpatentable over Drexler in view of U.S. Pat. No. 6,658,426 to Poskanzer (hereinafter "Poskanzer").

VII. ARGUMENTS

A.   Summary of the Applied Rejections

In the Final Office Action, the Examiner rejected claims 1-5, 10-12, 14-17, 22-24, 26-31, 36-38, 40-43, 48-50, 52-58, 64-65, and 67-90 under 35 U.S.C. §102(e) as being anticipated by Drexter (U.S. App. No. 2002/0046248). Claims 6-9, 32-35 and 59-63 were rejected under 35 U.S.C. §103(a) as being unpatentable over Drexler in view of Demers et al. (U.S. Patent No. 5,870,761). Claims 13 and 39 were rejected under 35 U.S.C. §103(a) as being unpatentable over Drexler in view of Huth et al. (U.S. Patent No. 6,704,742). Claims 18-21, 25, 44-47, 51, and 66 were rejected under 35 U.S.C. §103(a) as being unpatentable over Drexler in view of Poskanzer (U.S. Patent No. 6,658,426).

In so doing, the Examiner stated:

As to claim 1, Drexter teaches a method for converting messaging data into a relational table format in a database system, wherein the messaging data is within a messaging system (see page 1, paragraph 0002), the method comprising the steps of:
(a)   providing a table formatting specifications; (see page 2, paragraph 0029);
(b)   utilizing the plurality of table formatting specifications to automatically build and store a table function in the database system (see page 3, parapgraph 0034, where it is inherent that the associations (functions) are stored if they are going to be retrieved or recalled);
(c)   invoking the table function to access the messaging data (see pages 2-3, paragraphs 0030-0033); and
(d)   converting the messaging data by the table function into specific data types according to the plurality of table formatting specifications, wherein the

messaging data is transformed into the relational table format (see page 3, paragraph 0033).

. . .

As to claim 67, Drexter teaches a system for generating a customized invocation mechanism (see page 1, paragraph 0002), comprising:

an interface for receiving customizations (see page 3, paragraph 0034-0037); and

a software module coupled to the interface for building an invocation mechanism based on the customization specifications and storing the invocation mechanism in a database (see page 3, paragraph 0034, where it is inherent that the associations (functions) are stored if they are going to be retrieved or recalled), wherein the invocation mechanism is invokable by the database for accessing data external to the database (see page 3, paragraphs 0036-0037).

The Examiner further asserted that:

Drexler does disclose saving the function on the database system (see paragraph 0034, where it is inherent that if the association is going to be recalled at a later time it is saved). Drexter clearly discloses that one embodiment of the present invention is for it to be executed on a (single) computer system (see paragraph 0034). No where in Drexter is there evidence that the invention requires multiple systems or that parts of the invention would be executed on different systems. . . . It is noted that a database is simply a storage area, and that a database system is needed to invoke any program, application, or mechanism, and it is therefore assumed that the applicant is referring to the database system when reciting the limitation "wherein the invocation mechanism is invokable by the database" in claims 67, 75, and 83.

B.   The Cited Prior Art

1.   Drexler

Drexler is directed to an application program module that retrieves a message, parses the message into strings (when appropriate), and then applies a database association to the parsed strings in order to store the strings in fields in a database. (FIG. 2; ¶0029-¶0033). The database association is configured by the user (¶¶ 0034-0036; see FIG. 4) and stored in a file system in the computer system (¶0041). By checking a box, the user can indicate that messages pertaining to a particular database association be automatically retrieved periodically (¶0042).

2.   Demers et al.

Demers is directed to a method and system for duplicating at a destination site changes made to data at a source site. According to Demers, a plurality of streams are established between the source site and the destination site. The plurality of streams are used in parallel to propagate changes made at the source site to the destination site. A record of transactions that made changes that need to be propagated from the source site to the destination site is maintained at the source site. Before propagating changes made by a transaction to the destination site, the record of transactions is inspected to identify a set of transactions whose changes are not known to have been made permanent at the destination site. It is then determined whether the transaction could possibly depend on any transaction in the set of transactions. If the transaction could not possibly depend on any transaction in the set of transactions, then the changes made by the transaction are propagated to the destination site using one of the plurality of streams. (Abstract).

3.   Huth et al.

In Huth, a method and apparatus is provided for arranging and accessing database data in a manner such that massive amounts of data can be aggregated and manipulated in many different ways to generate reports of many different types in a rapid manner. In Huth, the method includes storing data in point slices where each slice includes data having similar attributes, receiving a report request from which data attributes corresponding to the data needed to instantiate the report can be gleaned, identifying at least one required point slice including the needed data, determining if the point slice exists, where the point slice does not exist, accessing other data and generating the point slice and perhaps some intervening point slices, storing the

newly generated point slices and then using the required point slice to instantiate and provide the report. (Abstract).

4. Poskanzer

Poskanzer is directed to an interface that provides a level of abstraction between the structure of a database system and application programs which access that system. The database system is represented by a model comprised of objects which correspond to the components of the database system. An object at a higher level encapsulates information contained in these other objects regarding the structure of the database. Whenever an application program requires access to the database, it sends a message to the higher level encapsulation object. The lower-level objects implement methods which automatically generate appropriate database commands. When the encapsulation object receives a call from an application program requesting data in the database, it instructs table objects to obtain the required data. In response, the table objects invoke field objects to identify how to represent data in each of the database fields to which they correspond. The table object concatenates the responses received from each of the field objects to construct a command that is presented to the database to retrieve the desired data. (Abstract).

C. Claims 1-5, 10-12, 14-17, 22-24, 26-31, 36-38, 40-43, 48-50, 52-58, 64-65 and 67-90 Are Allowable Over Drexler.

Appellants respectfully submit that Drexler fails to teach or suggest the present invention, as recited in claims 1, 27, 53, 67, 75 and 83. In particular, Drexler does not teach or suggest storing "a table function in the database system," as recited in claims 1, 27 and 53, or "storing the invocation mechanism in a database," as recited in claims 67, 75 and 83. In the present invention, the table function and invocation mechanism is stored in the database or database

system (the terms are interchangeable). This allows the table function and invocation mechanism to be invoked via an SQL statement understood by the database/database system. Accordingly, a separate application module is not required to use the table function and invocation mechanism.

In Drexler, FIG. 1 shows that the Email to Database Import Program 40, the database association 60, and the database 80 are separate components (¶¶0025-0027). There is no teaching or suggestion that the association 60 is stored in the database/database system, as recited in claims 1, 27, 53, 67, 75 and 83. In fact, Drexler explicitly states that the associations 60 can be found in "memory files, such as those on a floppy diskette, on the computer's hard drive, or a network hard drive." (¶0041).

In the Final Office Action and in the Advisory Action, the Examiner states that the claims are to be given their broadest reasonable interpretation in light of the supporting disclosure. In so doing, the Examiner states that the association is inherently stored because they can be used repeatedly. Appellants do not dispute this. The Examiner also states that the association is probably stored on the same computer system as the database/database system. Appellants do not dispute this either. Nevertheless, the Examiner contends that the database/database system and the computer system are one and the same and that therefore, because the association is stored in the computer system, it is also stored in the database system. Appellants strongly disagree.

Contrary to the Examiner's position, a database is not "simply a storage area." It is well known in the industry and in the art that a "database" is defined as a set of related files that is created and managed by a database management system (DBMS). (See http://www.techweb.com/encyclopedia). According to another definition,

[a] database is a collection of data elements (facts) stored in a computer in a systematic way, such that a computer program can consult it to answer questions. . . . The computer program used to manage and query a database is known as a database management system (DBMS). . . . Strictly speaking, the "database" is the collection of facts and the software is the "database management system" or DBMS. However, in practice, many database administrators and programmers use the term "database" to cover both meanings.

(See http://en.wikipedia.org/wiki/Database). Drexler explicitly states that the database 80 "may be a commercially available or privately created program . . . . The database 80 preferably includes a number of records, tables and/or fields to which data may be recorded." (¶0027). It is well known that while the database/database system can reside in a single computer system, such as a server, only the "collection of data elements" stored in the database/database system are managed by the database/database system. Other files, information and data stored in the file system of the computer system, outside of the database/database system, are managed by the operating system or some other application(s) in the computer system, and not by the database/database system.

Thus, while Appellants acknowledge that the claims are to be given their broadest reasonable interpretation, that interpretation must be *reasonable* and *reasonable in light of the supporting disclosure*. In the Advisory Action, the Examiner's interpretation of what constitutes a database or database system is "the system that includes the *hardware* that performs the action of storing data; the instructions, software, or programs running on the hardware that cause it to perform the action of storing data; and the hardware that actually does the data storing." (Advisory Action). Appellants respectfully submit that that interpretation is not *reasonable* because it defies all definitions of a database or database system, including that definition provided in Drexler itself. In addition, the Examiner's interpretation is not *reasonable in light of*

the supporting disclosure because the Specification explicitly is directed to a database management system (FIG. 1, item 80) coupled to a storage device 60 that resides in a computer system (FIG. 1, item 10c) along with a message queue 30. The table function, which is a UDF 70, is stored in the database 80 (Specification, page 11, lines 19-21).

Appellants respectfully submit that storing Drexler's associations in a memory file in the computer system or in a storage device of the computer system is not equivalent to storing the associations in the database/database system. Accordingly, Appellants respectfully submit that Drexler fails to teach or suggest storing "a table function in the database system," as recited in claims 1, 27 and 53, or "storing the invocation mechanism in a database," as recited in claims 67, 75 and 83.

In addition, Drexler also fails to teach or suggest "invoking the table function from within the database system," as recited in claims 1, 27 and 53, or an "invocation mechanism [that] is invokable by the database," as recited in claims 67, 75 and 83. In the present invention, the table function can be invoked from *within* the database/database system via an SQL statement. Because the database/database system understands the table function, it can invoke and execute it.

In Drexler, the Import Program 40 has access to the association 60 and to the database 80. According to Drexler, the "utility program 40 uses the association 60 to associate and save certain data from the email message 10 to appropriate records, tables or fields in the database 80. (¶ 0028). Thus, the utility program 40, and not the database, invokes the association. Indeed, Drexler's database is passive, e.g., the database only receives and stores data strings in fields. The database does not have the power to invoke or the ability to understand the association.

In the Final Office Action, the Examiner asserts that Drexler teaches "reusing the association (function) using the name given to the association (see paragraph 0034) which is read on invoking the application from within the database system." Again, the Examiner contends that the database system and the computer system are one and the same. Based on the discussion above, Appellants respectfully submit that while the database system can reside in the computer system, the computer system and the database system are not one and the same. Thus, using a utility application program in the computer system to invoke the association is not equivalent to invoking the association "from within the database system," as recited in claims 1, 27 and 53, and having an association that is invokable by the application program is not equivalent to an "invocation mechanism [that] is invokable by the database," as recited in claims 67, 75 and 83.

Finally, Drexler fails to teach or suggest "converting the messaging data . . . into specific data types, wherein the messaging data is transformed into the relational table format," as recited in claims 1, 27 and 53. In the present invention, the table function converts data from the message into specific data types according to the plurality of table formatting specifications so that the data can be transformed into a relational table format. The relational table format is a row with columns of desired data types. (Specification, page 10, lines 10-11). In Drexler, the association corresponding to a message associates parsed data strings with database fields (¶0033). The association can perform other functions that manipulate the parsed data so that it is consistent with other data stored in the database (¶¶ 0067). Nevertheless, nothing teaches or suggests that the association transforms the messaging data "into the relational table format," as recited in claims 1, 27 and 53.

Based on the foregoing, Appellants respectfully submit that Drexler fails to teach or suggest the present invention, as recited in claims 1, 27, 53, 67, 75 and 83. Accordingly those claims are allowable over Drexler. Claims 2-5, 10-12, 14-17, 22-24, 26, 28-31, 36-38, 40-43, 48-50, 52, 54-48, 64-65, 69-74, 76-82, and 84-90 depend from independent claims 1, 27, 53, 67, 75 and 83, respectively. Thus, claims 2-5, 10-12, 14-17, 22-24, 26, 28-31, 36-38, 40-43, 48-50, 52, 54-48, 64-65, 69-74, 76-82, and 84-9 are also allowable over Drexler.

D. Dependent claims 2, 28 and 54 Are Allowable Over Drexler for Additional and Alternative Reasons.

Appellants respectfully submit that claims 2, 28 and 54 are allowable over Drexler because they depend from claims 1, 27 and 53, respectively, and because Drexler fails to teach or suggest a table function that "invokes at least one messaging function within the database system," as recited in claims 2, 28 and 54. In the present invention, the table function and the messaging function are UDFs that are stored in the database system. The table function invokes the messaging function from within the database system to retrieve the messaging data.

In Drexler, the import utility program receives the email message and uses the association to associate parsed data from the email message to appropriate records, tables or fields in the database. (¶0028). Nothing in Drexler teaches or suggests that the association invokes the utility program from within the database system, as recited in claim 2. In the Final Office Action, the Examiner asserts that Drexler teaches this feature at ¶0042. That paragraph states that the user who is configuring the association can "enable automated data import by the association" by checking a box. By checking the box, the association is scheduled for an automated email to database import process, which means the utility import program will receive email periodically

and check to see if the received email should be processed by the association. Nothing in paragraph 0042 teaches or suggests that the association "invokes at least one messaging function within the database system," as recited in claims 2, 28 and 54.

Based on the above reasoning, Appellants respectfully submit that Drexler fails to teach or suggest the present invention, as recited in claims 2, 28 and 54.

E.   Dependent claims 3, 29 and 55 Are Allowable Over Drexler for Additional and Alternative Reasons.

Appellants respectfully submit that claims 3, 29 and 55 are allowable over Drexler because they depend from claims 1, 27 and 53, respectively, and because Drexler fails to teach or suggest that the table function and the messaging function "are user-defined functions within the database system," as recited in claims 3, 29 and 55. As stated above, the table function and the messaging functions are user-defined functions (UDFs) within the database system. UDFs are well known in the art. In general, a UDF is a routine that has been defined or programmed by the user of the system and has been included in a standard library of functions. In the context of a database system, the UDF is a set of SQL statements with an assigned name that is stored in the database so that it can be invoked by other UDFs or within an SQL statement. Nothing in Drexler teaches or suggests that the association or the import utility are "are user-defined functions within the database system," as recited in claims 3, 29 and 55.

In the Final Office Action, the Examiner asserts that Drexler teaches this at ¶0034. That paragraph, however, merely discusses FIG. 3 and how a user creates an association. Also the association is created by the user, it is not a "user-defined function within the database system." Also, there is not teaching or suggest in paragraph 0034 that the messaging function, e.g., the

import utility program, is also a UDF within the database system. Accordingly, Appellants respectfully submit that Drexler fails to teach or suggest the present invention, as recited in claims 3, 29 and 55.

F.   Dependent claims 6-9, 32-35 and 59-63 Are Allowable Over Drexler in view of Demers.

Claims 6-9, 32-35 and 59-63 depend from claims 1, 27, and 53, respectively. Thus, claims 6-9, 32-35 and 59-63 are also allowable over Drexler. Because Demers fails to remedy the deficiencies of Drexler, Appellants respectfully submit that dependent claims 6-9, 32-35 and 59-63 are allowable over Drexler in view of the Demers.

G.   Dependent claims 13 and 39 Are Allowable Over Drexler in view of Huth.

Claims 13 and 39 depend from claims 1 and 27, respectively. Thus, claims 13 and 39 are also allowable over Drexler. Because Huth fails to remedy the deficiencies of Drexler, Appellants respectfully submit that dependent claims 13 and 39 are allowable over Drexler in view of the Huth.
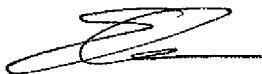
H.   Dependent claims 18-21, 25, 44-47, 51 and 66 Are Allowable Over Drexler in view of Poskanzer.

Claims 18-21, 25, 44-47, 51 and 66 depend from claims 1, 27, and 53, respectively. Thus, claims 18-21, 25, 44-47, 51 and 66 are also allowable over Drexler. Because Poskanzer fails to remedy the deficiencies of Drexler, Appellants respectfully submit that dependent claims 18-21, 25, 44-47, 51 and 66 are allowable over Drexler in view of the Poskanzer.

I. Summary of Arguments

For the reasons set forth above, Appellants respectfully submit that the claims 1-90 are

allowable over the cited references. Appellants respectfully request that the final rejection of

claims 1-90 be reversed.

Respectfully submitted,

SAWYER LAW GROUP LLP

Dated: ___April 6, 2006___

Erin C. Ming
Attorney for Appellant(s)
Reg. No. 47,797
(650) 475-1449

APPENDIX OF CLAIMS

1.   (Previously Presented) A method for converting messaging data into a relational table format in a database system, wherein the messaging data is within a messaging system, the method comprising the steps of:

(a)   providing a plurality of table formatting specifications;

(b)   utilizing the plurality of table formatting specifications to automatically build and store a table function in the database system;

(c)   invoking the table function from within the database system to access the messaging data; and

(d)   converting the messaging data by the table function into specific data types according to the plurality of table formatting specifications, wherein the messaging data is transformed into the relational table format.

2.   (Original) The method of claim 1, wherein the table function invokes at least one messaging function within the database system.

3.   (Original) The method of claim 2, wherein the table function and the at least one messaging function are user-defined functions within the database system.

4.   (Original) The method of claim 3, wherein the at least one messaging function retrieves and reads messaging data in the message system.

5.   (Original) The method of claim 1, wherein the providing step (a) further includes the step of:

       (a1)   reading the plurality of table formatting specifications from a file.

6.   (Original) The method of claim 1, wherein the providing step (a) further includes the steps of:

       (a1)   selecting a name and a type for the table function, wherein the type includes one of a retrieve function and a read function;

       (a2)   specifying where the table function is to be stored; and

       (a3)   indicating where the messaging data resides.

7.   (Original) The method of claim 6, wherein the specifying step (a2) further includes the steps of:

       (a2i)   providing a database name and access information; and

       (a2ii)   allowing the user to validate the access information.

8.   (Original) The method of claim 6, wherein the indicating step (a3) further includes the step of:

       (a3i)   providing a service point name for the messaging data.

9.   (Original) The method of claim 6, wherein the indicating step (a3) further includes the step of:

       (a3i)   providing a system default endpoint for the messaging data.

10.   (Original) The method of claim 1, wherein the providing step (a) further includes the step of:

(a1)   providing formatting information about the messaging data.

11.   (Previously Presented) The method of claim 10, wherein the providing step (a1) further includes the steps of:

(a1i)   designating a delimiter character, wherein the delimiter character separates the messaging data into column data.

12.   (Previously Presented) The method of claim 11, wherein the converting step (d) further comprising:

(d1)   invoking a parser function within the database system for parsing the delimited messaging data.

13.   (Previously Presented) The method of claim 12, wherein the invoking step (d1) further includes:

(d1i)   checking for the parser function within the database system;

(d1ii)   building the parser function if it does not exist within the database system; and

(d1iii) registering the parser function to the database system after it is built.

14.   (Original) The method of claim 10, wherein the providing step (a1) further includes the step of:

> (a1i)   specifying a fixed-length format by indicating a position and length of each column.

15.   (Original) The method of claim 10, wherein the providing step (a) further includes the step of:

> (a2)   allowing a user to view the messaging data in the messaging system to verify the formatting information provided.

16.   (Original) The method of claim 1, wherein the messaging data comprises a message string, the message string including a plurality of substrings, wherein each substring represents data that is returned as a column in a table.

17.   (Original) The method of claim 16, wherein the providing step (a) further includes the step of:

> (a1)   defining a column for each substring of the plurality of substrings in the message string.

18.   (Original) The method of claim 17, wherein the defining step (a1) further includes the steps of:

> (a1i)   naming each column; and

> (a1ii)   designating a data type for each column.

19.   (Original) The method of claim 18, wherein the defining step (a1) further includes the step of:

          (a1iii) allowing the user to view the messaging data formatted according to the column definitions provided.

20.   (Original) The method of claim 19, wherein the providing step (a) further includes the step of:

          (a2)   building the table function based on the table formatting specifications collected from the user.

21.   (Previously Presented) The method of claim 20, wherein the converting step (d) further includes:

          (d1)   parsing the message string into the plurality of substrings; and

          (d2)   converting each substring into the designated data type corresponding to its column.

22.   (Original) The method of claim 1, wherein the providing step (a) further includes the step of:

          (a1)   allowing a user to create and name a table view based on the table formatting specifications.

23.   (Previously Presented) The method of claim 22, wherein the invoking step (c) further includes the step of:

    (c1)   selecting messaging data from the table view.

24.   (Original) The method of claim 1, wherein the providing step (a) further includes the step of:

    (a1)   allowing a user to review a summary of the table formatting specifications before building the table function.

25.   (Previously Presented) The method of claim 3, wherein the invoking step (c) further includes the step of:

    (c1)   integrating the table function within a structured query language statement.

26.   (Previously Presented) The method of claim 4 further including populating directly a relational table in the database system with the returned messaging data.

27.   (Previously Presented) A computer readable medium containing programming instructions for converting messaging data into a relational table format in a database system, wherein the messaging data is within a messaging system, comprising the programming instructions for:

    (a)   providing a plurality of table formatting specifications;

    (b)   utilizing the plurality of table formatting specifications to automatically build and store a table function in the database system;

(c)   invoking the table function from within the database system to access the messaging data; and

(d)   converting the messaging data by the table function into specific data types according to the plurality of table formatting specifications, wherein the messaging data is transformed into the relational table format.

28.   (Original) The computer readable medium of claim 27, wherein the table function invokes at least one messaging function within the database system.

29.   (Original) The computer readable medium of claim 28, wherein the table function and the at least one messaging function are user-defined functions in the database system.

30.   (Original) The computer readable medium of claim 29, wherein the at least one messaging function retrieves and reads messaging data in the message system.

31.   (Original) The computer readable medium of claim 27, wherein the providing instruction (a) further includes the instruction for:

        (a1) reading the plurality of table formatting specifications from a file.

32.   (Original) The computer readable medium of claim 27, wherein the providing instruction (a) further includes the instructions for:

        (a1)   selecting a name and a type for the table function, wherein the type includes one of a retrieve function and a read function;

(a2)   specifying where the table function is to be stored; and

(a3)   indicating where the messaging data resides.

33.   (Original) The computer readable medium of claim 32, wherein the specifying instruction (a2) further includes the instructions for:

      (a2i)   providing a database name and access information; and

      (a2ii)   allowing the user to validate the access information.

34.   (Original) The computer readable medium of claim 32, wherein the indicating instruction (a3) further includes the instruction for:

      (a3i)   providing a service point name for the messaging data.

35.   (Original) The computer readable medium of claim 32, wherein the indicating instruction (a3) further includes the instruction for:

      (a3i)   providing a system default endpoint for the messaging data.

36.   (Original) The computer readable medium of claim 27, wherein the providing instruction (a) further includes the instruction for:

      (a1)   providing formatting information about the messaging data.

37.   (Original) The computer readable medium of claim 36, wherein the providing instruction (a1) further includes the instruction for:

      (a1i)   designating a delimiter character, wherein the delimiter character

separates the messaging data into column data.

38.   (Previously Presented) The computer readable medium of claim 37, wherein the

converting step (d) further comprising:

      (d1)   invoking a parser function within the database system for parsing the

delimited messaging data.

39.   (Previously Presented) The computer readable medium of claim 38, wherein the

invoking step (d1) further includes:

        (d1i)   checking for the parser function within the database system;

        (d1ii)   building the parser function if it does not exist within the database

system; and

        (d1iii) registering the parser function to the database system after it is

built.

40.   (Original) The computer readable medium of claim 36, wherein the providing instruction

(a1) further includes the instruction for:

        (a1i)   specifying a fixed-length format by indicating a position and length

of each column.

41.   (Original) The computer readable medium of claim 36, wherein the providing instruction

(a) further includes the instruction for:

        (a2)   allowing a user to view the messaging data in the messaging system to

verify the formatting information provided.

42.   (Original) The computer readable medium of claim 27, wherein the messaging data comprises a message string, the message string including a plurality of substrings, wherein each substring represents data that is returned as a column in a table.

43.   (Original) The computer readable medium of claim 42, wherein the providing instruction (a) further includes the instruction for:

 (a1)   defining a column for each substring of the plurality of substrings in the message string.

44.   (Original) The computer readable medium of claim 43, wherein the defining instruction (a1) further includes the instructions for:

 (a1i)   naming each column; and

 (a1ii)   designating a data type for each column.

45.   (Original) The computer readable medium of claim 44, wherein the defining instruction (a1) further includes the instruction for:

 (a1iii)   allowing the user to view the messaging data formatted according to the column definitions provided.

46. (Original) The computer readable medium of claim 45, wherein the providing instruction (a) further includes the instruction for:

  (a2) building the table function based on the plurality of table formatting specifications collected from the user.

47. (Previously Presented) The computer readable medium of claim 46, wherein the converting step (d) further includes:

  (d1) parsing the message string into the plurality of substrings; and

  (d2) converting each substring into the designated data type corresponding to its column.

48. (Original) The computer readable medium of claim 27, wherein the providing instruction (a) further includes the instruction for:

  (a1) allowing a user to create and name a table view based on the table formatting specifications.

49. (Previously Presented) The computer readable medium of claim 48, wherein the invoking instruction (c) further includes the instruction for:

  (c1) selecting messaging data from the table view.

50. (Original) The computer readable medium of claim 27, wherein the providing instruction (a) further includes the instruction for:

  (a1) allowing a user to review a summary of the table formatting specifications

before building the table function.

51.   (Previously Presented) The computer readable medium of claim 29, wherein the invoking instruction (c) further includes the instruction for:

(c1)   integrating the table function within a structured query language statement.

52.   (Previously Presented) The computer readable medium of claim 30 further including populating directly a relational table in the database system with the returned messaging data.

53.   (Previously Presented) A system for converting messaging data into a relational table format in a database system, wherein the messaging data is within a messaging system, the system comprising:

a processor;

a table function building application executable by the processor for receiving a plurality of table formatting specifications and for utilizing the plurality of table formatting specifications to automatically build and store a table function in the database system ; and

means for invoking the table function from within the database system to access the messaging data,

wherein, once invoked, the table function converts the messaging data into specific data types according to the plurality of table formatting specifications and transforms the messaging data into the relational table format.

54.   (Original) The system of claim 53, wherein the table function invokes at least one

messaging function within the database system.

55.   (Original) The system of claim 54, wherein the table function and the at least one

messaging function are user-defined functions within the database system.

56.   (Original) The system of claim 55, wherein the at least one messaging function retrieves

and reads messaging data in the message system.

57.   (Original) The system of claim 53, wherein the table function building application

includes a means for collecting the table formatting specifications from a user.

58.   (Original) The system of claim 53, wherein the table function building application

includes means for downloading the table formatting specifications from a file.

59.   (Original) The system of claim 57, wherein the collecting means comprises a graphical

user interface, wherein the graphical user interface prompts a user to select a name and a type for

the table function, wherein the type includes one of a retrieve function and a read function, to

specify where the table function is to be stored, and to indicate where the messaging data resides.

60.   (Original) The system of claim 59, wherein the graphical user interface further prompts

the user to provide formatting information about the messaging data.

61.   (Original) The system of claim 59, wherein the messaging data comprises a message

string, the message string including a plurality of substrings, wherein each substring represents

data that is returned as a column in a table.

62.   (Original) The system of claim 61, wherein the graphical user interface further allows the

user to define a column for each substring of the plurality of substrings in the message string.

63.   (Original) The system of claim 59, wherein the table function building application builds

the table function based on the plurality of table formatting specifications collected through the

graphical user interface.

64.   (Original) The system of claim 53, wherein the table function building application allows

a user to create and name a table view based on the plurality of table formatting specifications.

65.   (Original) The system of claim 64, wherein the invoking means includes means for

selecting messaging data from the table view.

66.   (Original) The system of claim 55, wherein the invoking means includes means for

integrating the table function within a structured query language statement.

67.   (Previously Presented) A system for generating a customized invocation mechanism,

comprising:

        an interface for receiving customizations; and

        a software module coupled to the interface for building an invocation mechanism based

on the customization specifications and storing the invocation mechanism in a database, wherein the invocation mechanism is invokable by the database for accessing data external to the database.

68.   (Previously Presented) The system of claim 67, wherein the invocation mechanism is dynamically generated.

69.   (Previously Presented) The system of claim 67, wherein the invocation mechanism further comprises at least one of the group consisting of: a UDF, a table function, a virtual table, a stored procedure, a trigger, a query statement, and a federated table, and an equivalent of any of the foregoing.

70.   (Previously Presented) The system of claim 67, further comprising means for invoking the invocation mechanism from a database.

71.   (Previously Presented) The system of claim 67, further comprising means for converting data accessed by the invocation mechanism into a format understood by the database.

72.   (Previously Presented) The system of claim 67, wherein the interface further comprising a graphical user interface for receiving function customization specifications.

73.   (Previously Presented) The system of claim 67, wherein the customization specifications further comprise specification of a relational format for nonrelational data accessed by the

customized function.

74. (Previously Presented) The system of claim 67, wherein the interface further comprises means for previewing nonrelational data in relational format based on customization specifications.

75. (Previously Presented) A method for generating a customized invocation mechanism, comprising the steps of:

receiving customization specifications; and

building an invocation mechanism based on the customization specifications and storing the invocation mechanism in a database, wherein the invocation mechanism is invokable by the database for accessing data external to the database.

76. (Previously Presented) The method of claim 75, wherein the invocation mechanism is dynamically generated.

77. (Previously Presented) The method of claim 75, wherein the invocation mechanism further comprises at least one of the group consisting of: a UDF, a table function, a virtual table, a stored procedure, a trigger, a query statement, and a federated table, and an equivalent of any of the foregoing.

78. (Previously Presented) The method of claim 75, further comprising the step of invoking the invocation mechanism from a database.

79.   (Previously Presented) The method of claim 75, further comprising the step of

converting data accessed by the invocation mechanism into a format understood by a database.

80.   (Previously Presented) The method of claim 75, wherein the function customization

specifications are received through a graphical user interface.

81.   (Previously Presented) The method of claim 75, wherein the customization specifications

further comprise specification of a relational format for nonrelational data accessed by the

customized function.

82.   (Previously Presented) The method of claim 75, further comprising the step of

previewing nonrelational data in relational format at a user interface based on user

customizations.

83.   (Previously Presented) A program product containing instructions executable by a

computer, the instructions embodying a method for generating a customized invocation

mechanism, comprising the steps of:

        receiving customization specifications; and

        building an invocation mechanism based on the customization specifications and storing

the invocation mechanism in a database, wherein the invocation mechanism is invokable by the

database for accessing data external to the database.

84.   (Previously Presented) The method of claim 83, wherein the invocation mechanism is

dynamically generated.

85.   (Previously Presented) The method of claim 83, wherein the invocation mechanism

further comprises at least one of the group consisting of: a UDF, a table function, a virtual table,

a stored procedure, a trigger, a query statement, and a federated table, and an equivalent of any of

the foregoing.

86.   (Previously Presented) The method of claim 83, further comprising the step of invoking

the invocation mechanism from a database.

87.   (Previously Presented) The method of claim 83, further comprising the step of

converting data accessed by the invocation mechanism into a format understood by a database.

88.   (Previously Presented) The method of claim 83, wherein the function customization

specifications are received through a graphical user interface.

89.   (Previously Presented) The method of claim 83, wherein the customization specifications

further comprise specification of a relational format for nonrelational data accessed by the

customized function.

90.   (Previously Presented) The method of claim 83, further comprising the step of

previewing nonrelational data in relational format at a user interface based on user

customizations.